FieldMaxII LabVIEW Examples – Getting Started

The standard FieldMaxII-PC software needs to be installed prior to using these LabVIEW examples since the USB driver for the meter is included as part of that installation. These VIs use the FieldMax2Lib.dll library for communication, which will also be installed on the computer automatically as part of the standard FieldMaxII-PC installation.

There are different copies of these LabVIEW examples included here. Each different copy contains the exact same VIs except that they have been saved specifically for different operating systems and different versions of LabVIEW. The configuration notes here can help in determining which copy of the examples to use…

For Windows 7 64-bit operating systems and 32-bit versions of LabVIEW

(If you are unsure of your version of LabVIEW (32-bit or 64-bit) please check the "Help > About LabVIEW" screen within your copy of LabVIEW.) The FieldMax2Lib.dll file will be installed into the "C:\Windows\SysWOW64" folder for this operating system. For that .dll location, the examples in the "FMII LV Examples – SysWOW64" folder should be used. Within that folder, there are different copies of the example VIs saved for different versions of LabVIEW. Select the copy of these examples that most closely matches the copy of LabVIEW you are using. (Each different copy of these examples is the same, but is just saved to match up with different versions of LabVIEW.)

For Windows 7 64-bit operating systems and 64-bit versions of LabVIEW

(If you are unsure of your version of LabVIEW (32-bit or 64-bit) please check the "Help > About LabVIEW" screen within your copy of LabVIEW.) A new copy of the FieldMax2Lib.dll had to be created specifically for building programs in 64-bit versions of LabVIEW. Because of this, you will need to manually copy this new version of the .dll into the System32 folder on your computer. You will find the new .dll in the following location on the computer:
C:\Program Files (x86)\Coherent\FieldMaxII PC\Drivers\Win7_x64
Copy the "FieldMax2Lib.dll" file out of that folder and paste it into the "C:\Windows\System32" folder on the computer. For that .dll location, the examples in the "FMII LV Examples – System32" folder should be used. Within that folder, there are different copies of the example VIs saved for different versions of LabVIEW. Select the copy of these examples that most closely matches the copy of LabVIEW you are using. (Each different copy of these examples is the same, but is just saved to match up with different versions of LabVIEW.)

For Windows XP and Windows 7 32-bit operating systems

The FieldMax2Lib.dll file will be installed into the "C:\Windows\System32" folder for 32-bit operating systems. For that .dll location, the examples in the "FMII LV Examples – System32" folder should be used. Within that folder, there are different copies of the example VIs saved for different versions of LabVIEW. Select the copy of these examples that most closely matches the copy of LabVIEW you are using. (Each different copy of these examples is the same, but is just saved to match up with different versions of LabVIEW.)

Example VI Descriptions

These LabVIEW examples include a handful of VIs that can be used to establish a connection with the FieldMaxII, collect data from it, and control the settings on the meter. Here is a brief description of the different VIs that are included…

Connect to Meter.vi

This VI needs to be run first to establish communication with the USB port on the FieldMaxII meter. The default Meter Index will be "0" if only one meter is connected. The Meter Index should be incremented for each additional meter that is connected. When this VI is run, it will return a Meter ID value and a Meter Serial Number. The Meter ID value should be wired through to each additional VI to tell the VI which meter to communicate with. If this VI does not connect to the meter properly, check to make sure the “Location of the FieldMax2Lib.dll” control matches the actual location of that .dll on the computer. If this still does not connect to the meter, try running the “Monitor Connection” VI prior to this one to help determine the Meter Index value.

Disconnect From Meter.vi

This VI should be run at the end of the program to close communications with the meter. Avoiding this step at the end of a program or routine could cause LabVIEW to crash unexpectedly.

Collect Data.vi

This VI is used to collect measurement data from the meter's buffer. The Measurement value in this VI will represent a reading in either Watts or Joules based on what measurement mode the meter is set for. The Measurement value here has not been formatted to match the exact format of what is being shown on the meter’s display. In general, additional measurement digits shown in this VI are not necessarily valid since the resolution of the measurement should be limited to what is shown on the meter’s display.

The Period value is only used with energy measurements and can be converted to the rep rate at which pulses are being collected by the meter.

1 / (Period x 8 usec) = Rep Rate

If the Measurement data appears to not be matching up with the data on the meter's display, doing a Re-Sync should clear the meter's buffer and synchronize the readings again. This Re-Sync may only be needed after changing a setting on the meter (like the power range it is operating in) or if there is some other disruption with collecting data while the program is running.

When using a power sensor with the FieldMaxII (like the models PM10, PM30, PS10, OP-2, etc), the meter will always update its readings at 10 Hz. This VI is set to run at 20 Hz to be sure that the measurement buffer is being cleared properly so that new readings are always being read into the VI. If measurements are being made with a pyroelectric energy sensor (like the models J-25MB-LE, J-50MB-YAG, etc), the meter will always update at whatever rate the laser pulses are being fired (up to a maximum of 300 Hz.) So the timing of this loop may need to be adjusted if running at higher rep rates than the 20 Hz this loop is currently set for.

Zero Meter.vi

This VI will zero the meter. Zeroing is only relevant with power sensors (not energy sensors) and is typically recommended prior to starting a new batch of power measurements. Since zeroing the meter can take different lengths of time (depending on the sensor and range settings being used), this VI will check for the status of the zero process and will run on a loop until the zero process is complete.

Meter Settings.vi

This VI is used to send commands and receive a reply from the meter for controlling the settings on the FieldMaxII. The commands for each setting can be found in the "Meter Settings Commands" document that is included with these VIs. Each different setting on the meter (like wavelength and range settings) will have a command associated with it. The commands are made up of a short text string along with an extension that can either be more text or a numeric value. Sending the command alone (without any extension) will query the meter for the setting related to that command. A full list of the commands and the settings they control are included in the “Meter Settings Commands” document.

Example – Stream Measurement.vi

This is a simple example to show a very basic routine for collecting data from the meter. This uses the “Collect Data” VI and adds the “Connect” and “Disconnect” VIs to it. So it connects to the meter, collects streaming data, then disconnects from the meter when stopped.

Example – Single Measurement.vi

This example for collecting data just returns a single measurement value instead of streaming measurements. The VI will automatically stop after the first valid measurement is collected. This can be useful if measurements are being taken on an interval instead of every data point being collected.

Example – Query Settings.vi

This example shows how the “Meter Settings” VI can be used to query the meter for some of its settings, including the wavelength setting, measurement mode, and currently selected range. The same type of sequence could be used to set these parameters into the meter as well.

Utility – Bytes to Values.vi

This VI is a utility used to convert raw data from the meter's buffer into measurement values. This utility will be needed within any VI that is used to collect data from the meter and is used in a couple of the VIs included here.

Monitor Connection.vi

This VI may or may not be needed depending on how the LabVIEW VIs are being used to interface with the FieldMaxII meter. This VI can be used to monitor meter connections on the computer. If a meter is added or removed from the computer, this VI will trigger an event that can be used to carry out additional functions if needed. If a meter is added to the computer, the “Add/Remove Event” should trigger and the “Index Found?” should trigger which indicates that a new Index value has updated in the VI. If a meter is removed from the computer, the “Add/Remove Event” should trigger, but no new Index value will be found or created. (These actions by the VI can be viewed by running the VI and then connecting and disconnecting the USB cable connected to the FieldMaxII meter.) If only one meter is connected to a computer, the meter’s index will typically be “0”. Then for each meter that is added to the computer after that, the index will increment. This index value is used in the “Connect To Meter” VI to establish a connection with the FieldMaxII. At that point, a Meter ID value is created which should be passed along to each additional VI that is used. The issue with just using the “Connect To Meter” VI, is that if a FieldMaxII is disconnected from the computer (or loses connection through some other means) while a VI is trying to communicate with the meter, it can cause LabVIEW to crash unexpectedly. If a FieldMaxII connection is lost with the computer, it does not gracefully disconnect from LabVIEW. That’s where this “Monitor Connection” VI can be useful. If a meter connection is lost, this will trigger an event which can be used to stop other VIs from running and could possibly be used to disconnect and reconnect to the meter if it becomes available to be connected to again. This type of setup becomes more complex since the “Monitor Connection” VI would need to be run in a different thread than any of the other VIs that are being used. Then if the “Monitor Connection” VI triggers an event, that event would need to be used to cause some action in the other VIs being used to avoid a crash in LabVIEW. This type of setup could be especially useful if LabVIEW is being used for long-term data collection from the FieldMaxII. For example, if the connection to the FieldMaxII is lost during data collection, the event created by this VI could be used to stop data collection and save all data up to that point instead of causing the whole program to crash and all data to be lost. If just quick connections are being done with the FieldMaxII (like a short data collection period), then often times just using the “Connect To Meter” VI with an appropriate Index Value can be used to connect to the meter and carry out what is needed. But if a long-term connection to the meter (and especially if long-term data collection is being used), then this “Monitor Connection” VI may be useful to help avoid any unexpected crashes by LabVIEW. This VI can also be useful if multiple meters are being connected to the computer since it generates index values for each meter that is discovered on the computer, which can help confirm what index values should be used in the “Connect To Meter” VI. This “Monitor Connection” VI monitors changes in meter connections. So it will only update with a new event or index if a meter has been added or removed from the computer. Then once a meter is found, it will not be found again unless the USB port is disconnected and reconnected. For this reason, it would be advised to not include this VI in a normal loop cycle. It will not find any new index values unless there is a physical change in the connection status of a meter. So if this is used, it needs to be run in a separate loop from other VIs that are being used to collect data or control settings on the FieldMaxII meter. At the end of this VI, there is a “DeInit” function that is added, which helps release all connections to the FieldMax2Lib.dll. This has been added to also help avoid errors in LabVIEW. This will cut all connections with the meter though and should not be run while other VIs are communicating with the meter. Once the DeInit function is called, the meter cannot communicate with other VIs until the “Connect To Meter” sequence is run again. The DeInit function is not valid with the .dll that is used for communication on XP operating systems. So if using this with Windows XP, the DeInit function should be deleted from this VI in order to avoid errors.